RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Duplicate Records Detection and Prioritization: A Case Study for a U.S. County

By Andrea Rozario
Of Rutgers, the State University of New Jersey
for the
36th WCARS
June 2, 2016
Sao Paulo, BR

Continuous Auditing

- Continuous auditing entails the real-time monitoring and analysis of the entire population of records (Vasarhelyi and Halper 1991)
- Premise of this methodology is based on the concept of audit- by-exception where deviations (e.g. control variances) are flagged as alerts and forwarded to the responsible parties (e.g. management, internal auditors, business owners) for investigation
- There is an increasing trend to follow an audit-by-exception approach
- Important to maintain a high level of quality of data in order to rely on the results of such approach

Why is the detection of duplicate records important?

- Business and governmental entities generate a substantial amount of data every day
- This data is used to perform analyses that can support decision making:
  - Using prior year purchasing data as a baseline to create an expenditure budget
  - Assuring the quality of the CAFR (Comprehensive Annual Financial Report)
- Important to ensure the quality of the data that is generated by an entity’s relational database
- Shortage of studies that address the problem of duplicate records in the governmental accounting literature
- CA literature is rich with studies that propose statistical and machine learning techniques to identify exceptions, but the results of duplicate records detection are usually too many (Dull et al., 2006; Kogan et al., 1999)

What is the issue with identifying too many duplicates?

Solution to duplicate record detection problem

- How can we devise a methodology to rank the detected duplicates in order to enable the human users to focus their attention on the more suspicious cases?

Duplicate records

Costly Problem

Causes:

- Different formats, structures or schema of databases
- Lack of a global or unique identifier
- Human factors (data entry, lack of constraints, intentional)

Detection Methods:

1. Exact matching:

   Records are *identical*

2. Fuzzy (near-identical) matching (Weis et.al., 2008):

   - Records have *similar* values for certain relevant fields
   - Causes: data entry errors, different value formats, etc. E.g. 10/21/10 vs. October 21, 2010
   - Classified as duplicates based on a threshold and some similarity criteria

Vendor Name	Address
J.B. Smith	1 Washington Park
J. Smith	1 Washington Park
John Smith	1 Washington Park Ave
John Smith	1 Washington Park Avenue

Duplicate Detection Process

Generalized framework (Weis & Neumann, 2005):

- Phase 1: Candidate definition *(offline)*
  - Determine which objects to compare

- Phase 2: Duplicate definition *(offline)*
  - Determine criteria (description + similarity measure) to use in order to consider actual duplicates

- Phase 3: Actual duplicate detection
  - Specifying how to detect duplicates candidates and find which ones are true duplicates

Record	Vendor Name	Address	Age	Phone
1	John Smith	1 Washington Park	32 yrs	973-123-4567
2	J.B. Smith	1 Washington Park	32 years	1-973-123-4567
3	J. Smith	1 Washington Park	32 years	(973)1234567
4	John Smith	1 Washington Park Ave	32 years	+1-973-123-4567
5	John Smith	1 Washington Park Avenue	32 yrs	+19731234567

Data

Data Description

1 file: (August 2011 – June 2015)

- **Dataset:** information on payments to various vendors; 473,000 records, 230 variables

Software & Algorithm used

Excel (data cleaning and preparation)

IDEA (duplicates detection)

Algorithm: 3-way match (Payee + Invoice Date + Invoice Amount)

- Additional variable: Invoice number

Algorithms and Findings

Dataset

- (Date, Amount, Vendor) yielded 83,000 candidates
- (Date, Amount, Vendor, Invoice ID) yielded 8,000 candidates

Duplicate Candidates Prioritization

- Large numbers of candidates
- Use a set of criteria to differentiate (rank) between them
- Simply adding a new variable to the algorithm proved suboptimal

Proposed prioritization based on a Composite Score:

$$CS_i = \sum W_{iCr_j}$$

Where CS_i is the Composite Score of the set of duplicate candidates *i*

W_{iCr_j} is the weight of Criterion j when applied to the set of duplicate candidates *i*

Proposed set of criteria:

Materiality, missing values, count of similar candidates, frequency per user, frequency per vendor, duplicate invoice number

Prioritization Criteria

- **Materiality:** $W_{i\_Materiality} = (Amt_i)/(\sum Amt_i)$
- **Missing values:** $W_{i\_MissValue} =$

$$\begin{cases} 1/(\sum Count_i), & if \text{ the set of duplicate candidates } i \text{ does not have missing values} \\ 0, & Otherwise \end{cases}$$

- **Count of similar candidates:** $W_{i\_Count} = (Count_i)/(\sum Count_i)$
- **Frequency per user:** $W_{i\_FreqUser} = (Count_{U_j i})/(\sum Count_i)$
- **Frequency per vendor:** $W_{i\_FreqVndr} = (Count_{V_j i})/(\sum Count_i)$
- **Duplicate invoice number:** $W_{i\_InvID} =$

$$\begin{cases} 1/(\sum Count_i), & if\ the\ Invoice\ ID\ is\ the\ same\ for\ the\ candidates \\ 0, & Otherwise \end{cases}$$

Prioritization Example

Record #	Vendor ID	Invoice #	Date	$ Amount	Created by
1001	619505	1241225	5/11/2009	268.55	JDoe
2034	619505	1241225	5/11/2009	268.55	JDoe
9418	619505	1241225	5/11/2009	268.55	JDoe
7430	203339		7/7/2009	4119.5	JSmith
6159	203339		7/7/2009	4119.5	JSmith
8332	552751	1325148	10/5/2009	80.35	JDoe
4723	552751	1279869	10/5/2009	80.35	JDoe

For Record 1001 we calculate the following weights:

- $W_{1001\_Materiality} = (Amt_{1001})/(\sum Amt_i)$ =268.55/ 9205.35 = 0.0292
- $W_{1001\_MissValue} = 1/\ (\sum Count_i) =\ 1/7 = 0.1429$ (as there are no missing values causing it to be a duplicate candidate)
- $W_{1001\_Count} = (Count_{1001})/(\sum Count_i) = 3/7 =$ 0.4286
- $W_{1001\_FreqUser} = (Count_{U_j i})/(\sum Count_i) = 5/7 = 0.7143$
- $W_{1001\_FreqVndr} = (Count_{V_j i})/(\sum Count_i) = 3/7 = 0.$ 4286
- $W_{1001\_InvID} = 1/\ (\sum Count_i) = 1/7 = 0.1429$ (Invoice ID are the same)

CS$_{1001}$=1.8863

Ranking of the example

Composite Scores of all the duplicate candidates in the example:

Record #	Score - Materiality	Score - Missing Values	Score - Count	Score - Frequency by User	Score - Frequency by Vendor	Score - Invoice ID	Composite Score	Rank
1001	0.0292	0.1429	0.4286	0.7143	0.4286	0.1429	**1.8863**	1
2034	0.0292	0.1429	0.4286	0.7143	0.4286	0.1429	**1.8863**	1
9418	0.0292	0.1429	0.4286	0.7143	0.4286	0.1429	**1.8863**	1
7430	0.4475	0.0000	0.2857	0.2857	0.5714	0.0000	**1.5904**	4
6159	0.4475	0.0000	0.2857	0.2857	0.5714	0.0000	**1.5904**	4
8332	0.0087	0.1429	0.2857	0.7143	0.5714	0.0000	**1.7230**	6
4723	0.0087	0.1429	0.2857	0.7143	0.5714	0.0000	**1.7230**	6

Conclusion

- Given the recent emphasis on transparency and accountability of government funds, it is important to ensure the data is accurate and reliable
- In this study, we detected duplicate candidates for a U.S. county and proposed a prioritization framework to rank these candidates
- Next step: Apply the prioritization framework to the government data and refine the framework as we obtain feedback

рахмат
danke
謝謝
ngiyabonga
spas
Баярлалаа
mersi
kia ora
barka
welalin
tack
teşekkür ederim
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
gracias
хвала
kiitos
dankie
nanni
nandri
dhanyavad
mauruuru
köszönöm
thank you
dankon
ačiū
akun
asante
manana
obrigada
murakoze
tenki
bedankt
enkosi
bayarlalaa
gràçie
hvala
dziękuję
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
sobodi
děkuji
obrigado
mèsi
didi madloba
sagolun
sukriya
kop khun krap
taiku
ありがとう
najis tuke
tanemirt
grazie
arigatô
takk
dakujem
trugarez
kam sah hamnida
rahmat
terima kasih
rahmet
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci